ASSEMBLY X86

Di seguito il codice Assembly per la CPU x86 con una descrizione per ogni riga di codice:

0x00001141 <+8>: mov EAX,0x20 ; Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX

0x00001148 <+15>: mov EDX,0x38 ; Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX

0x00001155 <+28>: add EAX,EDX ; Somma il valore di EDX al valore di EAX e memorizza il risultato in EAX

0x00001157 <+30>: mov EBP,EAX ; Copia il valore del registro EAX nel registro EBP

0x0000115a <+33>: cmp EBP,0xa ; Confronta il valore del registro EBP con il valore esadecimale 0xa (10 in decimale)

0x0000115e <+37>: jge 0x1176 <main+61> ; Salta all'istruzione all'indirizzo 0x1176 se il valore di EBP è maggiore o uguale a 10

0x0000116a <+49>: mov eax,0x0 ; Carica il valore 0 nel registro EAX

0x0000116f <+54>: call 0x1030 <printf@plt> ; Chiama la funzione printf (il PLT è la Procedura Linkage Table, che gestisce le chiamate a funzioni esterne come printf)

Ecco una descrizione dettagliata di ogni istruzione:

1. mov EAX,0x20: Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. mov EDX,0x38: Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. add EAX,EDX: Somma il valore del registro EDX (56) al valore del registro EAX (32), memorizzando il risultato (88) in EAX.
4. mov EBP,EAX: Copia il valore del registro EAX (88) nel registro EBP.

5. cmp EBP,0xa: Confronta il valore del registro EBP (88) con il valore esadecimale 0xa (10 in decimale).
6. jge 0x1176 <main+61>: Salta all'indirizzo 0x1176 se il valore del registro EBP (88) è maggiore o uguale a 10.
7. mov eax,0x0: Carica il valore 0 nel registro EAX.
8. call 0x1030 <printf@plt>: Chiama la funzione printf tramite l'indirizzo memorizzato nella Procedura Linkage Table (PLT).